

Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Technologies



Diploma Thesis

**iOS-based communication and notification system of a
working group**

Author: Bc. Beisen Sultanmakmut

Supervisor: doc Ing. Vojtěch Merunka, Ph.D.

© 2018 CULS Prague

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Sultanmakhmut Beisen

Informatics

Thesis title

iOS-based communication and notification system of a working group

Objectives of thesis

The goal of this thesis is requirement analysis, design, implementation and testing an information system supporting communication of a small group. This software will use iOS and iPhone technology. Select the best implementation environment and realize this goal.

Methodology

The functional requirement model will be created first. Then, a conceptual analysis of a data model and workflow processes will be designed. Final implementation will be in some standard software framework Xcode. Following software engineering standards will be used: UML and Ambler's approach to the SW development, which will be in Objective-C or Swift programming language.

The proposed extent of the thesis

50-60 pages

Keywords

iOS, web, communication, notification, working group

Recommended information sources

Fitzek Frank H. P., Reichert Frank – Mobile phone programming and its application to wireless networking
ISBN: 978-1-4020-5968-1

Karmouch Ahmed, Korba Larry, Madeira, Edmundo R. M.- Mobility aware technologies and applications :
first international workshop, ISBN: 3-540-23423-3

Peter van de Put – Professional iOS Programming – ISBN: 978-1-118-66113-0

Stephen G. Kochan – Programming in Objective-C 2.0 – ISBN: 978-0-321-56615-7

Expected date of thesis defence

2017/18 SS – FEM

The Diploma Thesis Supervisor

doc. Ing. Vojtěch Merunka, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 21. 2. 2017

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 21. 2. 2017

Ing. Martin Pelikán, Ph.D.

Dean

Prague on 03. 03. 2018

Declaration

I declare that I have worked on my diploma thesis titled "iOS – based communication and notification system of working group" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break copyrights of any their person.

In Prague on 31.03.2018

Beisen Sultanmakhmut

Acknowledgement

I would like to thank doc. Ing. Vojtěch Merunka, Ph.D. for his professional advises and support during my work on this thesis. Furthermore, I want to thank Ing. Martin Kozák for his consultations and point of view on the topic of my Diploma Thesis.

iOS – based communication and notification system of a working group

Abstract

The topic of this diploma project is “Creating an iOS application called ‘Attention’”. Thesis was determined from the practical need to share urgent information in current city, to aware other users of the application.

The project consists of an explanatory statement containing 60 pages of text, including 12 figures, 5 formulas and 7 tables.

The paper contains: an abstract, an introduction, 4 chapters, a references and conclusion.

The introduction defines the actuality of the chosen topic, formulated the purpose and objectives of the study, indicated objects and subjects of the study.

In this graduation paper an application users can immediately post urgent situation in a city, to aware and endanger other users of the application.

Conclusion contains the key findings and trends that are designed to improve the quality of the project.

Keywords: mobile application, sharing information tool, urgent information, iOS, web, communication, notification, working group.

Komunikační a notifikační systém pro pracovní skupinu založený na systému iOS

Abstrakt

Tématem této diplomové práce je vytvoření aplikace na systému iOS pro pracovní skupinu. Práce vychází z praktické potřeby sdílet naléhavé informace na jednom místě tak, aby byli informováni všichni ostatní uživatelé této aplikace.

Projekt se skládá z popisu dokumentace obsahující 60 stran textu včetně 12 schémat, 5 vzorců a 7 tabulek.

Text obsahuje: abstrakt, úvod, 4 kapitol, odkazy a závěr.

V úvodu je vymezena aktuálnost zvoleného tématu, formuluje účel a cíle práce a označen předmět zkoumání.

Uživatelé aplikace mohou okamžitě vysílat zprávy například o naléhavou situaci v dané komunitě, uvědomit a upozornit ostatní uživatele aplikace.

Závěr obsahuje klíčové poznatky a trendy, které byly vybrány a použity pro zlepšení kvality projektu.

Klíčová slova: mobilní aplikace, sdílený informační nástroj, urgentní informace, iOS, web, komunikace, notifikace, pracovní skupina

Table of content

1. Introduction.....	11
2. Objectives and Methodology	12
2.1. Objectives	12
2.2. Methodology.....	12
3. Literature Review	13
3.1. Analysis of modern approaches.....	13
3.1.1. Features of mobile application.....	13
3.1.2. Analysis of the market.....	17
3.1.3. Features of applied tools.....	20
3.1.4. Statement of the problem.....	23
3.2. Development of models	23
3.2.1. Application server	23
3.2.2. Mobile software.....	25
3.2.3. Realization	25
3.3. Architecture of the distributed applications	26
3.3.1. Data presentation level.....	28
3.3.2. Data storage level	28
3.4. Concept of realization.....	29
3.4.1. Application server kernel	30
3.4.2. Value of this opportunity	30
3.5. Application servers: pluses and minuses	32
3.5.1. Advantages.....	32
3.5.2. Disadvantages	33
3.6. Development tools	33
3.6.1. Development for iOS	33
3.6.2. Xcode.....	34
3.6.3. Components.....	35
4. Practical Part.....	36
4.1. Architecture of application.....	36
4.2. The implementation of the application.....	37
4.2.1. Service.....	37
4.2.2. Server Part	39
4.2.3. Application.....	40
4.3. The interface design	41

5. Results and Discussion	45
5.1. Economic Efficiency	45
5.2. App Store (iOS).....	45
5.3. Calculation of the costs of developing software.....	46
5.3.1. Calculation of the cost of materials	46
5.3.2. Expenditure of salaries	47
5.3.3. The costs of operation and maintenance of equipment.....	48
5.3.4. Calculation of profit	49
5.4. Labor protection and Industrial Ecology.....	50
5.4.1. General information about health and labor protection.....	50
5.4.2. Analysis of dangerous and harmful factors.....	50
5.4.3. Measures of norms	50
5.4.4. Elimination of static loadings and monotony of work.....	52
5.4.5. Lighting.....	53
5.4.6. Electrical safety.....	53
5.4.7. Fire safety	54
5.4.8. Safety.....	55
5.4.9. Human impact on the environment of the object and activities for environmental safety.....	55
6. Conclusion	57
7. References.....	58
Appendix.....	59
List of helpful sources for developing iOS Applications.....	59
Most downloaded mobile applications on iOS.....	60

List of figures

Figure 1. Level of penetration of mobile devices.....	13
Figure 2. An example of the principal page of application.....	17
Figure 3. Incidents part.....	18
Figure 4. Mapy.cz application map.....	19
Figure 5. Application shows the destination time	20
Figure 6. “Application server” model	24
Figure 7. Use Case Diagram.....	37
Figure 8. Relation in WCF	38
Figure 9. Logo design	42
Figure 10. Interface of page “Main Page”	43

Figure 11. Interface page “Product”44
 Figure 12. Top Non-Game Apps by Revenue.....60

List of tables

Table 1. Comparative characteristic of operating systems21
 Table 2. Main Table.....39
 Table 3. List of methods.....41
 Table 4. Staffing of the project team.....47
 Table 5. The calculation of the basic and additional wages47
 Table 6. The estimated total costs of the work.....49
 Table 7. Microclimate parameters.....51

List of abbreviations

- NS - Foundation (OS X and iOS) and Application Kit (OS X)
- UI – User Interface Kit (iOS)
- AB – Address book
- CA – Core animation
- CI – Core image
- SVG – Scalable Vector Graphics
- PDA – Personal Digital Assistant
- ISO – International Organization for Standardization
- API – Application Programming Interface
- OS – Operational Systems
- MVC – Model-View-Controller
- VCS – Version Control System
- GUI – Graphical User Interface
- VAT – Value-Added Tax
- UI – User Interface
- VC – View Controller

1. Introduction

Today we are at the stage of technology depended age, which means, now we can not even imagine our everyday life without them. Personal computers, smart phones and Internet are parts of our life and their role is only increasing.

Especially, mobile technologies are increasingly developing for society helping terms. There is an infinite numbers of mobile applications, which are doing our lives simpler and more effective.

The main goal of diploma project was a development of mobile application, which helps users to share urgent information with other users of an application.

The main steps of diploma project were analysis of current market, development of the models and realization of the concept of application. Then there were the steps of implementation of architecture of application and implementation of design of this mobile application.

The level of developing of mobile device is increasing in developing countries. The level of penetration of mobile technologies is very high in developing and developed countries. Today situation is vivid example of that mobile devices have become a part of the life of a modern person.

The purpose of the diploma is creation an application of means of modern technologies, development of a mobile application on the basis of iPhone Operating System (iOS) is considered.

The level of penetration of mobile devices differs in different countries. The schedule in which it is visually possible to see how mobile devices get into an everyday life of the person. Proceeding from this it is possible to draw conclusions that mobile devices have become a part of the life of a modern person. The idea of application programming is very popular. Operating systems and infrastructure of mobile devices have been developed since the beginning of 2000.

2. Objectives and Methodology

2.1. Objectives

The goal of this thesis is requirement analysis, design, implementation and testing an information system supporting communication of a small group. This software will use iOS and iPhone technology. Select the best implementation environment and realize this goal.

2.2. Methodology

The functional requirement model will be created first. Then, a conceptual analysis of a data model and workflow processes will be designed. Final implementation will be in some standard software framework Xcode. Following software engineering standards will be used: UML and Ambler's approach to the SW development, which will be in Objective-C or Swift programming language.

3. Literature Review

3.1. Analysis of modern approaches

3.1.1. Features of mobile application

Modern technologies create a large number of tools convenient for daily using, it means that a person comes from the past, for example, they would seem to thirty-year prescription supernatural and even the alien.

The level of penetration of mobile devices differs in different countries (Figure 1). The schedule in which it is visually possible to see how mobile devices get into an everyday life of the person is given [Mobile Technology SDFTMM, 2014].

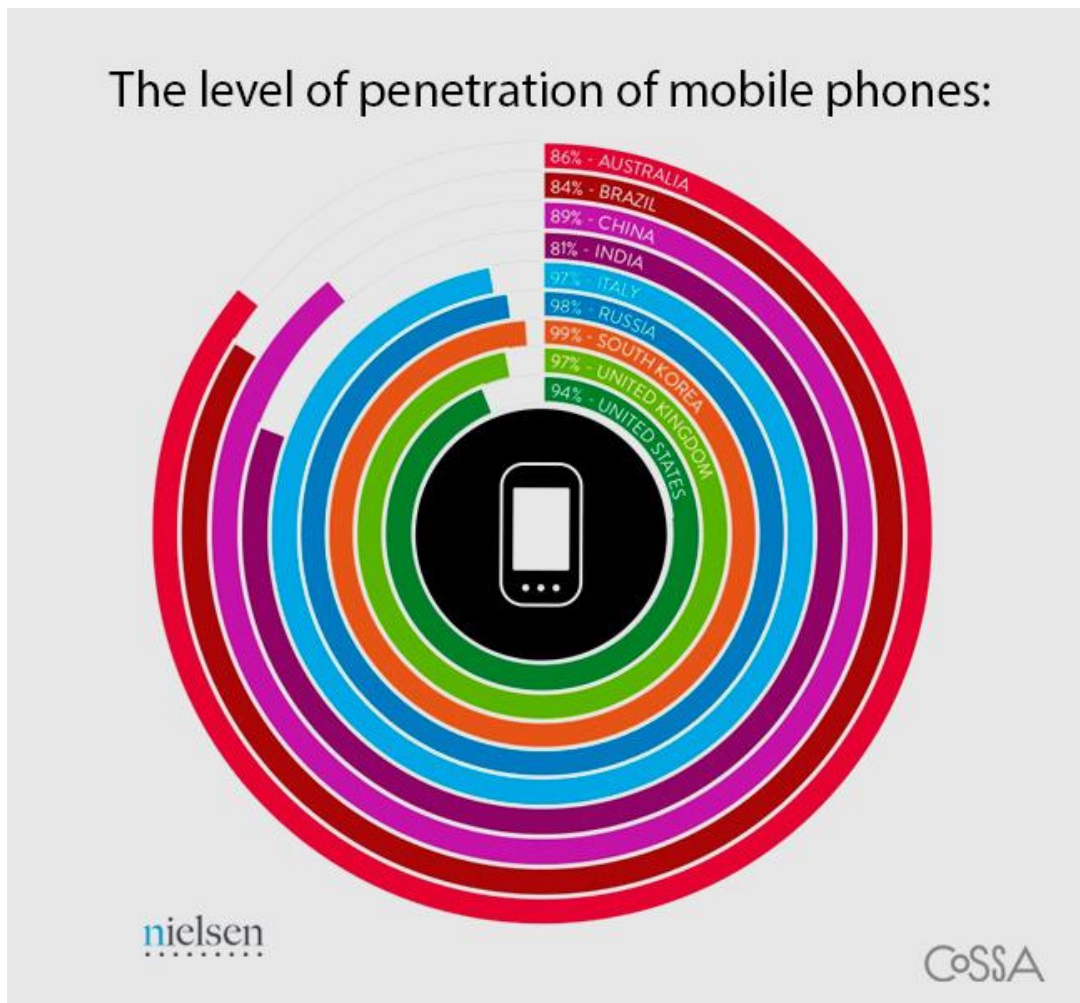


Figure 1. Level of penetration of mobile devices
Source: <https://wearesocial.com/uk/blog/2013/03/niensens-mobile-consumer-report>

Transmission of thoughts, words and images to distance for the 100-th fraction of a second is such a common reality, not everyone can imagine lives without these opportunities.

The unlimited access to the information in 5 years made changes, on what hundred years of natural development and evolution were not capable.

And separate independent phenomena was mobile applications. These are small programs, which are capable to create the whole certain world within one compact electronic device.

The rapid pace of technical change and growth in the mobile market. He placed strategic planning for the non-standard project before the developers not only from a technical point of view, but also because the share of the smartphone market is changing rapidly between different systems. Until recently, the iPhone iOS dominated the mobile market, but Google Android has now demonstrably overtaken iPhone in terms of market share, due partly to the power of the Google brand and partly to the platform's openness. Other mobile operating systems include the Blackberry RIM OS and Windows 8. As well as the wealth of mobile platforms emerging, there are now more hardware manufacturers than ever producing mobile devices. When considering how best to incorporate mobile technology into an existing business model, the primary issue for both clients and developers is currently the choice between native apps and web applications.

Other issues include:

- The inherent restrictions caused by loading, caching and latency issues on mobile devices.
- The unique interaction models in mobile hardware.
- The lack of adequate data transfer levels within mobile networks at this time.

We also need to consider that “mobile” no longer just means phone handsets; a range of other, new device types have grown in popularity, such as the tablet computer or other devices positioned somewhere between a lightweight laptop and a smartphone, and with some of the characteristics of both. The growth in mobile technologies has

meant that businesses in certain sectors are even receiving most of their web traffic from users browsing in mobile contexts.

Apps targeted at specific mobile platforms enjoy a number of natural benefits: Native access to the user interface creates a level of interaction that is difficult, if not impossible, to achieve through a Web browser. Native apps are in prime position to exploit the unique hardware and software facilities within mobile devices, such as GPS and localization tools, accelerometers and touchscreens. The primary consideration when implementing a service using mobile apps is the number of platforms. If targeting a sizeable chunk of the market is necessary, the resources required may be considerable.[Karmouch A. et al, 2004].

Mobile apps distributed commercially through app stores are subject to sales transaction charges. The task of promotion and adoption by new users is also increasingly challenging, as many of the app stores, particularly Android, are becoming extremely overcrowded. For some purposes, a native app may be primarily used as a marketing resource, providing a supplementary service which highlights some larger branding or commercial objective. Both short and long term native app development implications include. A diverse skill-set is necessary to develop apps for multiple platforms. There are significant maintenance implications, as the various operating systems, software and hardware contexts are in a constant state of flux. Prospective clients should therefore look for a demonstrably high level of competence when considering any development services for mobile. A wealth of experience puts Blueberry in a strong position to deliver not only development expertise but sound advice on these issues.

Speed. Mobile Web browsing has changed dramatically over the past couple of years and is predicted to change drastically again over the next two years, largely as a result of 4G.3 Time-frame is therefore an important consideration when planning a mobile project, as the mobile Web market will be totally different once 4G is widely available. Moreover, technological changes in the coming years will be substantial. Much has already changed. For example, today the screen size of a user's

mobile device may be anywhere between the size of the current smaller smartphones all the way up to the size of a small laptop computer, where a tablet device is being used.

An increase in bandwidth and network efficiency will mean speeds at least three times 3G services, with additional improvements suggesting a five-fold increase within the next decade. However, until the widespread deployment of 4G, reliance on Web applications alone is likely to be insufficient for many purposes.

Existing studies of mobile Web service users indicate that speed is valued over functionality, with users frequently frustrated in their attempts to use such services either quickly or successfully at all. In fact, statistics revealed through a study by Compuware's web performance division Gomez show that over 80% of users have not been able to complete a task such as a purchase, while around half have abandoned a site because of this problem. In general, over a third of users said they would begin abandoning a site within 5 seconds if it loads too slowly. [Mobile Technology SDFMM, 2014]. For software developers, the dominant issues to consider when undertaking a project for the mobile market are:

The pace of change in the mobile market makes strategic planning very difficult. A flexible position must be adopted to provide mobile services while keeping future options as open as possible. While future network technologies will vastly increase the potential of both native and Web apps, for the present applications must necessarily be lightweight. For the immediate future, both Web applications and native apps will continue to have significant relevance.

For prospective clients, the idea of being able to develop a Web application once and deploy it in multiple user environments is an appealing one. However, the pace of change in the mobile market makes strategic planning very difficult. With end users exploiting multiple environments and communicating through an ever-increasing range of channels, the mobile Web seems most likely to be in a position to reach all of them in the long term. However, given its present limitations, focusing exclusively on

the Web will remain challenging. The accepted view is that most of the inherent advantages of native apps will be accessible from within Web applications eventually.

3.1.2. Analysis of the market

These part of diploma work will show current market and classification of the application. There is no similar application in market, that is why we will consider to analyse remotely alike products. Decided to pick one feature from one field of an application.

1) The first comparative application is one of the most popular news application “Novinky.cz” (Figure 2).



Figure 2. An example of the principal page of application

Source: <https://www.mobileaction.co/app/ios/us/novinky-cz/430326937>

Popular newsfeed in Czech Republic, covers spheres like: Economy, Politics, Incidents, Industry, Culture, Life, Sport, Studies, Opinions etc.

The most relative to developing application is “Incidents” part in Novinky.cz application (Figure 3). It covers all urgent incidents not only in Czech Republic, but worldwide. Correspondent posts news 24/7.



Figure 3. Incidents part

Source: <https://itunes.apple.com/gb/app/novinky-cz/id430326937?mt=8>

In conclusion of the first comparative application, the application is effective tool to review all news in the World, however it does not covers local, really necessary and

productive information.

2) The second comparative application is “Mapy.cz”

Click anywhere on the map and you have a planned trip to the most interesting places in the area. You will see restaurants, ATMs, stops, accommodations and other places that can be useful. Plan your route by car, by bicycle, on foot, in summer also by water and cross-country skiing. (Figure 4).



Figure 4. Mapy.cz application map

Source: <https://itunes.apple.com/cz/app/mapy-cz/id411411020?mt=8>

Mapy.cz shows traffic jams in city, and best way to abandon traffic jams (Figure 5).

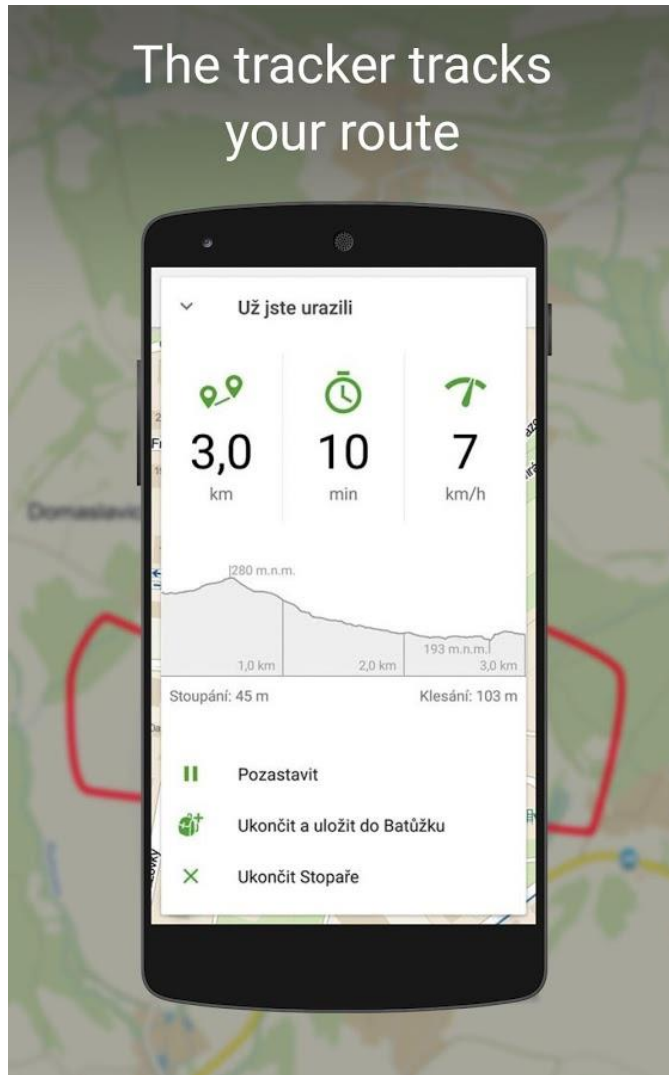


Figure 5. Application shows the destination time

Source: <https://play.google.com/store/apps/details?id=cz.seznam.mapy>

3.1.3. Features of applied tools

Some comparative proves of choosing iOS application development, instead Android application development.

Simple and effective. The new design of Apple's iOS 11 looks elegant and beautiful and its sleek and simple interface means that even a novice can use the iPhone without difficulty. iOS is all about 'less is more'. By taking away seldom used features more emphasis is placed on the important tasks. Yes, Android allows more customization, but iOS delivers more functionality. Less customization is less chance

for messing up your mobile device since most people actually use specific functions and don't want to know everything.

Multitasking. People say Apple's version of multitasking (when you use one program, the others stay paused) is not good but they actually forget that Android's version of multitasking uses up the battery very quickly. Moreover, a mobile device isn't made for multitasking in the same way as a desktop anyway.

Security. Unless it's been jail broken, iOS doesn't have any malware or bloatware that makes your mobile device slow and ugly. It has a far more stable system than Android because iOS is only used for iDevices. In other words, it was created to fit iDevices with the best optimization. Plus, iOS has a native Find My iPhone app, which allows iOS users to track the location of their device on another iDevice or a computer. With this app, your device is protected and your information is secure.

Fragmentation. Android is available on many different devices running different hardware specs, screen sizes, and versions of Android. Apple devices continue to update to the latest OS until the hardware simply can't handle anymore. However, updating Android phone devices is not easy work. The reason is that hardware manufacturers of Android phones and OS developers are different entities and there isn't the same fluid communication between hardware and OS as in the case of the iPhone. Apps are therefore a problem with Android. Some apps don't update and require older operating systems while some are new and don't support older versions. Likewise apps aren't capable of running on some systems because of the lack of hardware requirements.

App Quality. iOS has the best apps collection both in terms of quantity and quality. Any time a new app is released it is on iOS several months before it is on Android. iOS is also a highly polished OS – noticeable once you have used the same app on both systems. iOS apps are of much higher quality than the same app on Android [Fitzek Frank H.P., Reichert Frank, 2007].

OS	iOS	Android
Developer	Apple Inc.	Google

OS Family	OS X, UNIX	Linux
Customizability	Limited unless jailbroken	A lot. Can change almost anything
Programming language	C, C++, Objective-C	C, C++, Java
Easy media transfer	with desktop application	depends on model
Source model	High	Average
Browser	Closed, with open source components	Open source
Open source	The iOS kernel is not open source but is based on the open-source Darwin OS.	Kernel, UI, and some standard apps
Call features supported	Auto-respond, call-back reminder, do not disturb mode	Auto-respond
Supported display resolutions	Without restrictions	WVGA 800x480 WXGA 1280x768 HD 1280x720
Software catalogue	App Store	Play Market
Support of hardware acceleration	+	+

Support of contactless payments of NFC	+	+
Multitask of the processor	+	+

Table 1. Comparative characteristic of operating systems

Source: Author

3.1.4. Statement of the problem

The statement of the problem is the most important part of project. Definition of the problem is, was and will be the most crucial part of all creations. That is why, decided to choose, problem of the lack of awareness. Very strange to have these kind of technologies and not to cover one of the most important social duty, sharing information.

3.2. Development of models

Evacuation of connected rationale presents to discrete level to designers extra adaptability in formation of the appropriated data frameworks. Arrangement and execution of projects on the gathering of the server lessens prerequisites to equipment of customers and diminishes issues of guaranteeing similarity in the heterogeneous system environment.

3.2.1. Application server

The application server — is the service program which provides access of clients to the applied programs which are carrying out on the server. The application server is usually allocated as an average link (Figure 6) in three-level client-server architecture (3-tier).

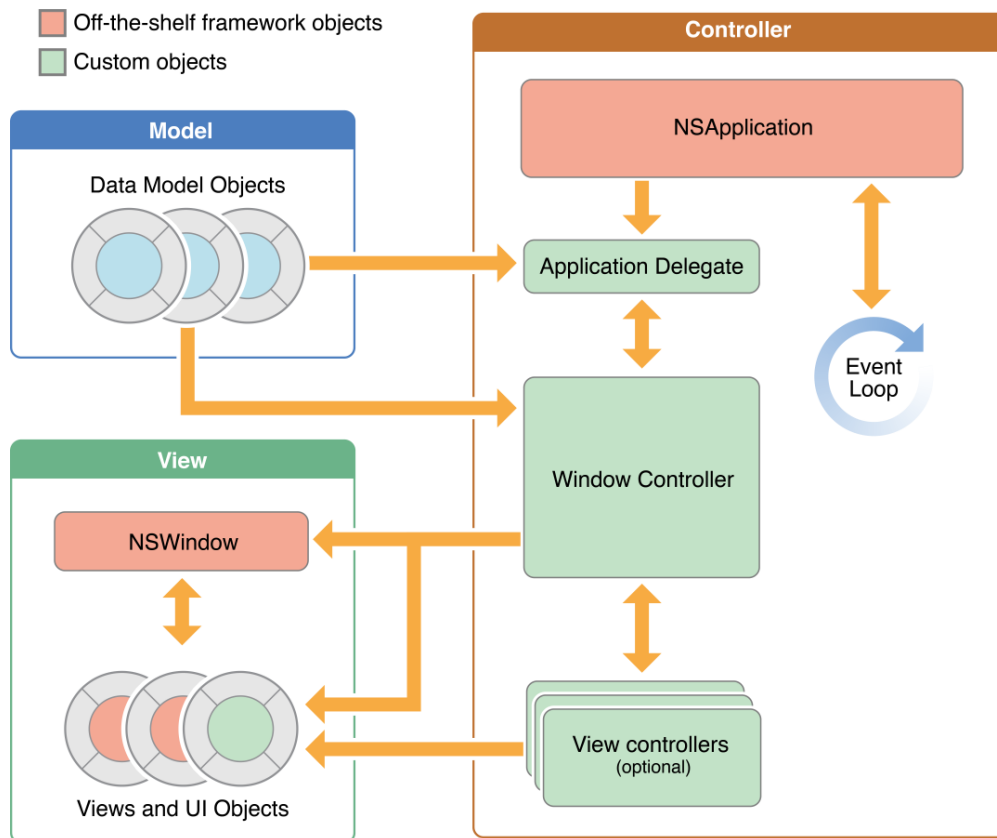


Figure 6. "Application server" model

Source: <https://developer.apple.com/library/content/documentation/General/Conceptual/MOSXAppProgrammingGuide/CoreAppDesign/CoreAppDesign.html>

Regardless of whether you are using a single-window or multiwindow app style, all apps use the same core set of objects. Cocoa provides the default behaviour for most of these objects. You are expected to provide a certain amount of customization of these objects to implement your app's custom behavior.

Figure 6 shows the relationships among the core objects for the single-window app styles. The objects in this figure are separated according to whether they are part of the model, view, or controller portions of the app. As you can see from the figure, the Cocoa-provided objects provide much of the controller and view layer for your app.

3.2.2. Mobile software

The style of the application characterizes which center articles must use in its usage. Cocoa underpins the production of both single-window and multiwindow applications. For multiwindow outlines, it likewise gives a report structural engineering to assist deal with the documents connected with every application window. Hence, applications can have the accompanying structures:

- Single-window utility application
- Single-window library-style application
- Multiwindow report based application

Pick a fundamental application style at a very early stage in outline process because that decision influences all that to do later. The single-window styles are favored as a rule, particularly for engineers bringing applications from iOS. The single-window style commonly yields a more streamlined client experience, and it additionally makes it simpler for application to bolster a full-screen mode. Nevertheless, application meets expectations broadly with complex records, the multiwindow style may be best on the grounds that it gives more archive related foundation to help you execute your application.

Utility applications regularly handle fleeting information or oversee framework forms. Mini-computer does not make or manage any reports or persevering client information yet basically forms numerical information entered by the client into the content field in its single window, showing the aftereffects of its computations in the same field. At the point when the client stops the application, the information it handled is essentially tossed.

3.2.3. Realization

Quick is intended to furnish consistent similarity with Cocoa and Objective-C. Utilize Objective-C APIs (extending from framework structures particular custom code) in Swift, and you can utilize Swift APIs in Objective-C. This similarity makes Swift a simple, helpful, and intense apparatus to coordinate into Cocoa application

advancement work process. This aide covers three critical parts of this similarity that use further bolstering good fortune when creating Cocoa applications:

- Interoperability interface in the middle of Swift and Objective-C code, permitting to utilize Swift classes in Objective-C and to exploit well known Cocoa classes, examples, and practices when composing Swift code.
- Blend and match permits to make blended dialect applications containing both Swift and Objective-C records that can correspond with one another.
- Relocation from existing Objective-C code to Swift is made simple with interoperability and blend and match, making it conceivable to supplant parts of Objective-C applications with the most recent Swift highlights.

3.3. Architecture of the distributed applications

Let's consider the architecture of the distributed applications allowing it to carry out difficult and various functions. Various options of creation of the distributed applications are given in different sources. And all of them have the right to existence, after all such applications solve the widest range of tasks in many subject domains, and uncontrollable development of development tools and technologies pushes to continuous improvement.

Nevertheless, there is most the general architecture of the distributed applications according to which it breaks into some logical layers, data processing levels. Applications, as we know, are intended for information processing, and here we can allocate their three major functions [Markov E, 2003]:

- data presentation (the user level). Here users of the applications can see necessary data, send a request for performance, enter new data into system or edit them;
- data processing (intermediate level, middleware). At this level the business logic of the application is concentrated, is controlled data flows and interaction of parts of the application will be organized. Concentration of all functions of data processing and management at one level is considered the main advantage of the distributed applications;

- data storage (level of data). It is level of servers of databases. Here servers, databases, accessors are located to data, various auxiliary tools.

Quite often such architecture call three-level or three-tier. And very often on the basis of these "three whales" the structure of the developed application is created. Thus it is always noted that each level can be in addition broken into some subtotals. For example, the user level can be broken into actually user interface and rules of check and processing of entered data.

Certainly, if to take into account possibility of splitting into subtotals, it is possible to enter any distributed application in three-level architecture. But here it is necessary to consider one more characteristic inherent in distributed applications, is a data control. Importance of this function is obvious as it is very difficult to create really working distributed application (with all client stations, middleware, DB servers etc.) which wouldn't operate the inquiries and answers. Therefore the distributed application has to possess one more logical level - data control level.

Therefore, it is expedient to divide an intermediate level into two independent: data processing level (as it is necessary to consider important advantage which it gives, - concentration of business rules of data processing) and data control level. The last provides control of performance of inquiries, serves work with data flows and will organize interaction of parts of system.

Thus, it is possible to allocate four main levels of the distributed architecture:

- data presentation (the user level);
- business logic rules (data processing level);
- data control (data control level);
- data storage (data storage level).

Three levels from four, excepting the first, are engaged directly in data processing, and level of data presentation allows to visualize and edit them. By means of this level users obtain data from data processing level which, in turn, takes information from storages and carries out all necessary transformations of data. After input of new information or editing existing data flows go to a way back: from the user interface through level of business rules in storage.

One more level - a data control - costs away from the main data flow, but it provides uninterrupted functioning of all system, operating inquiries and answers and interaction of parts of the application.

3.3.1. Data presentation level

Data presentation level - only available to the end user. This level models client workplaces of the distributed application and corresponding ON. Possibilities of a client workplace first of all are defined by opportunities of an operating system. Depending on type of the user interface the client software shares on two groups: the clients using possibilities of GIP (for example, Windows), and Web clients. In any case the client application has to provide performance of the following functions:

- data acquisition;
- data presentation for viewing by the user;
- editing of data;
- check of a correctness of the entered data;
- preservation of the made changes;
- processing of exclusive situations and display of information on mistake users.

It is desirable to concentrate all business rules at the level of data processing, but in practice it not always works well. Then speak about two types of client software. The "thin" client contains the minimum set of business rules, and "thick" realizes a considerable share of logic of the application. In the first case it is significantly easier to debug, modernize and expand the distributed application, in the second - it is possible to minimize expenses on creation and maintenance of level of a data control as the part of operations can be carried out on the party of the client, and on a share of middleware data transmission lays down only.

3.3.2. Data storage level

Once locked down your schema and class-level permissions, it's time to think about how data is accessed by your users. Object-level access control enables one user's

data to be kept separate from another's, because sometimes-different objects in a class need to be accessible by different people. For example, a user's private personal data should be accessible only to them.

Parse also supports the notion of anonymous users for those apps that want to store and protect user-specific data without requiring explicit login.

When a user logs into an app, they initiate a session with Parse. Through this session they can add and modify their own data but are prevented from modifying other users' data.

3.4. Concept of realization

Swift uses the same runtime as the existing Objective-C system but requires iOS 11 / OS X 10.12 or higher. Swift and Objective-C code can be used in a single program, and by extension, C and C++ as well. In contrast to C however, C++ code cannot be used directly from Swift. It is necessary to create an Objective-C or C wrapper between Swift and C++. In the case of Objective-C, Swift has considerable access to the object model, and can be used to subclass, extend and use Objective-C code to provide protocol support. The inverse is not true: Swift class cannot be subclass in Objective-C.

To aid development of such programs, and the re-use of existing code, Xcode 9 offers a semi-automated system that builds and maintains a "bridging header" to expose Objective-C code to Swift. This takes the form of an additional header file that simply defines or imports all of the Objective-C symbols that are needed by the project's Swift code. At that point, Swift can refer to the types, functions, and variables declared in those imports as though they were written in Swift.

Objective-C code can also use Swift code directly, by importing an automatically maintained header file with Objective-C declarations the project's Swift symbols. For instance, an Objective-C file in a mixed project called "MyApp" could access Swift classes or functions with the code `#import "MyApp-Swift.h"`. Not all symbols are available through this mechanism, however—use of Swift-specific features like generic types, non-object optional types, sophisticated enums, or even Unicode

identifiers may render a symbol inaccessible from Objective-C. [Stephen G.Kochan, 2008]

Swift also has limited support for attributes, metadata that is read by the development environment, and is not necessarily part of the compiled code. Like Objective-C, attributes use the @ syntax, but the currently available set is small. One example is the @IBOutlet attribute, which marks a given value in the code as an "outlet", available for use within Interface Builder (IB). An "outlet" is a device that binds the value of the on-screen display to an object in code.

3.4.1. Application server kernel

The promise of the new SDK is that these developers can now build a secure JavaScript application with a backend without having to provision or write code for servers, said James Yu, founder of San Francisco-based Parse.

With Parse, we handle user authentication, data storage to a server, etc. without the developer needing to make anything but a static HTML page. We believe this is the next inevitable step in application design for mobile web and traditional web apps.

The SDK builds atop DocumentCloud's popular Backbone.js, so those developers who know that will be able to adopt it easily to build HTML5 apps.

Parse — which already offers Android and iOS SDKs — competes with Stackmob but the real competition for the new JavaScript SDK is DIY server programming.

3.4.2. Value of this opportunity

Mediating controllers are typically ready-made objects drag from the Interface Builder library. Configure these objects to establish the bindings between properties of view objects and properties of the controller object, and then between those controller properties and specific properties of a model object.

As a result, when users change a value displayed in a view object, the new value is automatically communicated to a model object for storage—via the mediating

controller; and when a property of a model changes its value, that change is communicated to a view for display. The abstract `NSController` class and its concrete subclasses—`NSObjectController`, `NSArrayController`, `NSUserDefaultsController`, and `NSTreeController`—provide supporting features such as the ability to commit and discard changes and the management of selections and placeholder values.

A coordinating controller is typically an `NSWindowController` or `NSDocumentController` object (available only in `AppKit`), or an instance of a custom subclass of `NSObject`. Its role in an application is to oversee—or coordinate—the functioning of the entire application or of part of the application, such as the objects unarchived from a nib file. A coordinating controller provides services such as:

- Responding to delegation messages and observing notifications
- Responding to action messages
- Managing the life cycle of owned objects (for example, releasing them at the proper time)
- Establishing connections between objects and performing other set-up tasks

`NSWindowController` and `NSDocumentController` are classes that are part of the Cocoa architecture for document-based applications. Instances of these classes provide default implementations for several of the services listed above, and you can create subclasses of them to implement more application-specific behavior.

You can even use `NSWindowController` objects to manage windows in an application that is not based on the document architecture.

A coordinating controller frequently owns the objects archived in a nib file. As File's Owner, the coordinating controller is external to the objects in the nib file and manages those objects. These owned objects include mediating controllers as well as window objects and view objects. See *MVC as a Compound Design Pattern* for more on coordinating controllers as File's Owner.

Instances of custom `NSObject` subclasses can be entirely suitable as coordinating controllers. These kinds of controller objects combine both mediating and coordinating functions. For their mediating behavior, they make use of mechanisms such as target-action, outlets, delegation, and notifications to facilitate the movement of data between view objects and model objects.

They tend to contain a lot of glue code and, because that code is exclusively application-specific, they are the least reusable kind of object in an application.

3.5. Application servers: pluses and minuses

To see the whole picture advantages and disadvantages are considered.

3.5.1. Advantages

1) Integrity of a code and data. Business logic placement on the allocated server or limited number of server computers guarantees access to updated and modernized ON for all clients. It excludes risk of access and a data control from outdated and, probably, incompatible programs.

2) The centralized management. Changes in a configuration of applied programs, such as, for example, change of the server of databases, are carried out centrally.

3) Safety. The centralized means through which the service provider (service provider) can operate access to data and application components, allow to carry out check of authenticity of potentially unreliable clients on the average a layer and not to mention database level.

4) Productivity. The application server can solve problems of balancing of a network traffic and loading distribution between other physical servers of system.

5) Total cost of possession. Set of the listed above advantages, and in addition to them redistribution of expenses for the equipment with client on the server party, can lead to economy of means for the organization. As practice of rent of the software can be reflected in decrease in a total cost of possession. To be fair it should be noted that the cost of the server software, and also costs of its introduction and maintenance can be very high.

3.5.2. Disadvantages

1) Centralization. The systems constructed on the basis of an application server, have one main shortcoming inherent in all centralized decisions — "falling" of the server will result in inaccessibility of programs to all clients. Besides to effect will give also malfunctions in network connection.

2) Information security. This problem, in principle, is actual for any network decisions using for data transmission infrastructure of public networks.

3.6. Development tools

The iOS SDK (Software Development Kit) (formerly iPhone SDK) is a software development kit developed by Apple Inc to develop native applications for iOS.

3.6.1. Development for iOS

The Apple Ecosystem. Some developers and users prefer Apple's closed and more tightly controlled environment. Apple's ecosystem, consisting of Xcode, iCloud, iTunes, the App Store, the iPad, the iPhone, Mac computers, etc., offers a very holistic user experience with the ability to share data seamlessly across devices plus the content prior to going on sale.

Xcode. Apple's all-in-one IDE offers state-of-the-art offers features like a graphical interface builder and a syntax-aware and compiler-aware source editor.

The Apple Experience. From the fit and finish of its hardware to the customer service in its retail stores, to the way its hardware and software work flawlessly together, Apple has created the premium brand in consumer electronics. iOS developers share the benefits — so when someone downloads your iOS app, they already expect it's a great product. And that it will also work flawlessly on their device. [Peter van de Put, 2013]

Tablets. iOS has a clear advantage over Android in terms of tablet market adoption. Using Xcode, it's also much easier to migrate an app from a phone to a

tablet and vice versa than it is when developing for Android. Android OS SDK includes the following components:

- A Mac computer running OS X
- Xcode
- iOS SDK

Xcode is an integrated development environment (IDE) containing a suite of software development tools developed by Apple for developing software for OS X and iOS. First released in 2003, the latest stable release is version 6.3 and is available via the Mac App Store free of charge for OS X Yosemite users. Registered developers can download preview releases and previous versions of the suite through the Apple Developer website. However, Apple recently made a beta version of version 9.x of the software available to those of the public with Apple Developer accounts.

3.6.2. Xcode

Previously Xcode supported distributing a product build process over multiple systems. One technology involved was called Shared Workgroup Build, which used the Bonjour protocol to automatically discover systems providing compiler services, and a modified version of the free software product distcc to facilitate the distribution of workloads. Earlier versions of Xcode provided a system called Dedicated Network Builds. These features are absent in the supported versions of Xcode.

Mach-O executable format, which allows for "fat binaries," containing code for multiple architectures, Xcode can build universal binaries, which allow software to run on both PowerPC and Intel-based (x86) platforms and that can include both 32-bit and 64-bit code for both architectures. Using the iOS SDK, Xcode can also be used to compile and debug applications for iOS that run on the ARM processor. Xcode also includes Apple's WebObjects tools and frameworks for building Java web applications and web services (previously sold as a separate product). As of Xcode 3.0, Apple dropped [Polyak-Braginskiy A.V, 2002]. WebObjects development inside Xcode; WOLips should be used instead. Xcode 3 still includes the WebObjects frameworks. Xcode includes the GUI tool Instruments, which runs atop DTrace, a dynamic tracing framework created by Sun Microsystems and released as part of OpenSolaris.

The main application of the suite is the integrated development environment (IDE), also named Xcode. The Xcode suite also includes most of Apple's developer documentation, and built-in Interface Builder, an application used to construct graphical user interfaces.

Up to Xcode 4.1, the Xcode suite included a modified version of the GNU Compiler Collection. In Xcode 3.1 up to Xcode 4.6.3, it included the `llvm-gcc` compiler, with front ends from the GNU Compiler Collection and a code generator based on LLVM (which originally stood for Low Level Virtual Machine - see the LLVM article for clarification on this). In Xcode 3.2 and later, it included a code generator based on Apple's LLVM Compiler, and the Clang Static Analyzer. Starting with Xcode 4.2, the Apple LLVM Compiler became the default compiler, Starting with Xcode 5.0, LLVM was the only compiler provided.

Up to Xcode 4.6.3, the Xcode suite used the GNU Debugger (GDB) as the back-end for the IDE's debugger. Starting with Xcode 4.3, the LLDB debugger was also provided; starting with Xcode 4.5 LLDB replaced GDB as the default back-end for the IDE's debugger. Starting with Xcode 5.0, GDB was no longer supplied.

Xcode supports C, C++, Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, Rez, and Swift source code with a variety of programming models, including but not limited to Cocoa, Carbon, and Java. Third parties have added support for GNU Pascal, Free Pascal, Ada, C#, Perl, and D.

3.6.3. Components

Xcode 9 support for iOS 11 SDK, with always support of OS X 10.12 Sierra SDK but not the support of OS X 10.13 High Sierra SDK. This latest was only included in the betas version. It also added an LLVM 64 bit compiler for iOS 11. Apple removed support for building Garbage Collected Cocoa binaries in Xcode 8.1.

4. Practical Part

In this section will be described detailed implementation of the application. There are descriptions of classes, methods, procedures, database tables, etc. Also, charts of application and description of code will be submitted.

4.1. Architecture of application

Figure 7 is Use Case Diagram, where you can clearly see all of the processes taking place between the Client, applications and servers.

Client. The client, while using the application makes requests to the next, previous, and saying to a list of popular products. On the main page there is also a function of estimation products. The customer clicks on the button «Like» if he likes product. Thus there is a request for calculation on Application side. There is also another function - date function is where client makes a request for products on the entry.

Next, the application to commit to any request by product, processes and sends it to the server. The same thing happens with the date products. Application processing request waiting and further sends the request to the server. When using the «Like» there is a request by incrementing some products on the server.

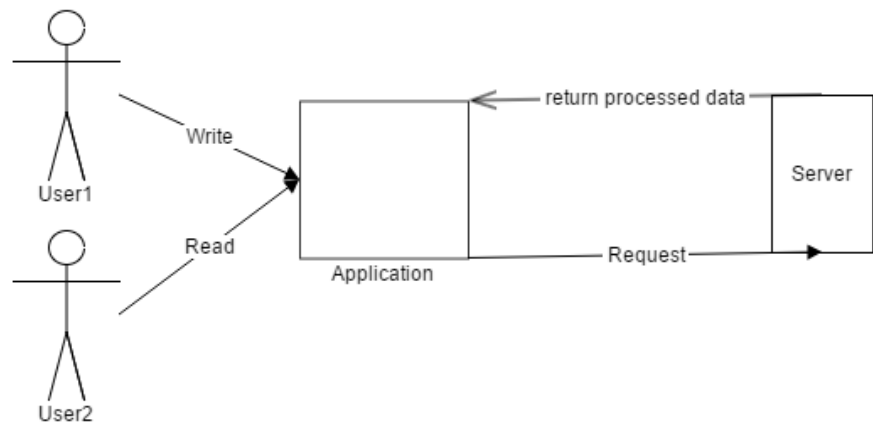


Figure 7. Use Case Diagram

Source: Author

The main function of the server is to return responses to requests from applications. Since all the products and the data associated with them, such as the id and the number of like-s are stored in a database on the server.

4.2. The implementation of the application

Here we are going to consider the implementation of the application.

4.2.1. Service

As stated in the application architecture, the work was divided into three parts. The main connecting link in is a Parse service. Parse service function is to process all the requests coming from applications and sending these requests to the server.

Actually the architecture based on the use of services, where different teams can write the different services, even on different technologies - a clear description of what a service can interact with another service - is a must.

Parse server contains two classes - Sender and Receiver. Parse technology is built on the concept of contracts, by which we can describe the interaction between the services, message format, etc.

This technology offers us three different types of contracts: the contract messages, data and services [Patrice Pelland, 2011] (Figure 8).

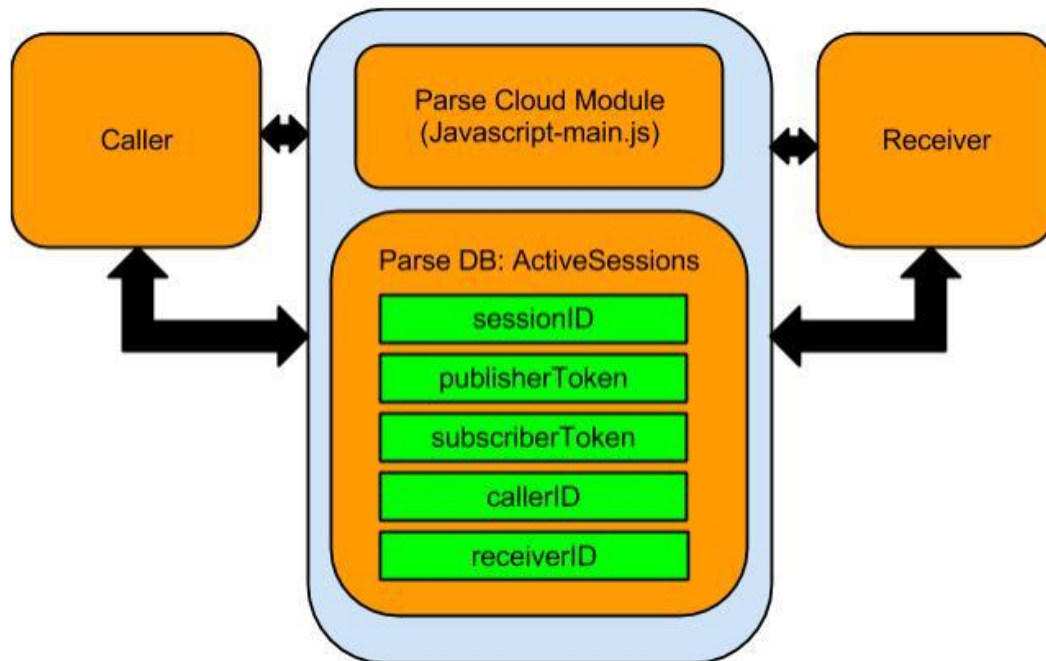


Figure 8. Relation in WCF

Source: <http://www.binpress.com/tutorial/creating-iphone-video-chat-app-using-parse-and-opentok-tokbox/22>

Service contract is a contract that specifies the direction and type of the messages in a conversation. A service contract is the gateway to a service for external applications to make use of the service functions, and at least one service contract should be available in a service. In our project is IService1.cs Service contract.

An operation contract defines the methods of the service that are accessible by external systems. The OperationContract attribute needs to be applied for all these methods, these are also like web methods in a web service. So, we list all the methods from the application. Operation Contracts that were used in this class:

- userID - String
- userLocation - Geopoint
- userTitle - String

The data contract - specifies the data format which service will be shared. This applies both to the request for service, and to the service's active. If you are using primitive types, like int, string, etc. the contract is not necessary, because the.

Net understands how to serialize and deserialize types, but if you use more complex types, such as Customers, Order, etc. You need to say how to serialize and deserialize these objects. DataMembers in this class:

- BoolValue
- StringValue

Next, we will examine the class of the service "Service1.svc". In this class we use TableAdapters to provide communication between our application and database by executing SQL statements and store procedures against a database.

4.2.2. Server Part

The main task of the server part in this thesis project is to store themselves products and data associated with it. To do this we created a table that will store the products and its attributes. The script to create the table can be studied in Application "A".

Description of table is given in Table 2 and DataSet is given in Figure 9.

Name	Data Type	Allow Nulls	Default
ObjectID	Str		
Name	Str		
Roles	Relation(Role)		
CreatedAt	Date		
UpdatedAt	Date		
ACL	ACL		

Table 2. Main Table

Source: Author

1) Attribute "ObjectId" is set as primary key because each product will be uniquely identified in table. As we count the rating of each product we need to identify each of them. Type of data is "str" to be able to count the rating.

2) name - contains the name of the user, so the type of data is set of chars. It may be empty, to avoid system errors, and it's allowed to have nulls

3) roles - contains the number of likes, pressed on each product. Data type is "int". Initially it will be 0, so it is also allowed to have nulls.

4) CreatedAt - contains date of creation account

5) UpdatedAt - contains date of updating an account of user

6) ACL - contains specific type of information for table ACL

4.2.3. Application

The main application class is ViewController.swift. All methods (table 3) and events, the connection to the Service is in this class. Next, we consider the details of these functions.

The main variables of the class ViewController.h:

- username - Username class object for the connection with the Service
- password - Password class object for the connection with the Service
- post - Posting class object for implementation of posting opportunity

Button Name	Function	Method
@IBOutlet	Sends request to email text field	func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool { Parse.setApplicationId("JzsV5QgSS5D7otX11YbQvslfX1nyg8k4nZgLcQan", clientKey: "0faoKKmq6Yb4WPvHCxWxbDeIcARL5HR7EEpUjRxe")
UIActivityIndicatorView	Activity Indicator	var actInd: UIActivityIndicatorView = UIActivityIndicatorView(frame: CGRectMake(0, 0, 150, 150)) as UIActivityIndicatorView
Alert	Alert	var alert = UIAlertView(title: "Invalid", message: "Username must be at least 4 characters and Password must be at least 5 symbols", delegate: self, cancelButtonTitle: "OK") alert.show()

Table 3. List of methods
(Author)

4.3. The interface design

Branding is certainly not a light topic – whole publications & hundreds of books have been written on the topic, however to put it in a nutshell you could describe a ‘brand’ as an organization, service or product with a ‘personality’ that is shaped by the perceptions of the audience. On that note, it should also be stated that a designer cannot “make” a brand – only the audience can do this. A designer forms the

foundation of the brand. The fundamental idea and core concept behind having a 'corporate image' is that everything a company does, everything it owns and everything it produces should reflect the values and aims of the business as a whole.



Figure 9. Logo design

Source: Author

There was developed two main pages:

- 1) "Main page" - One of the main pages, Here is the posts section: The user can read posts by chronological queue.

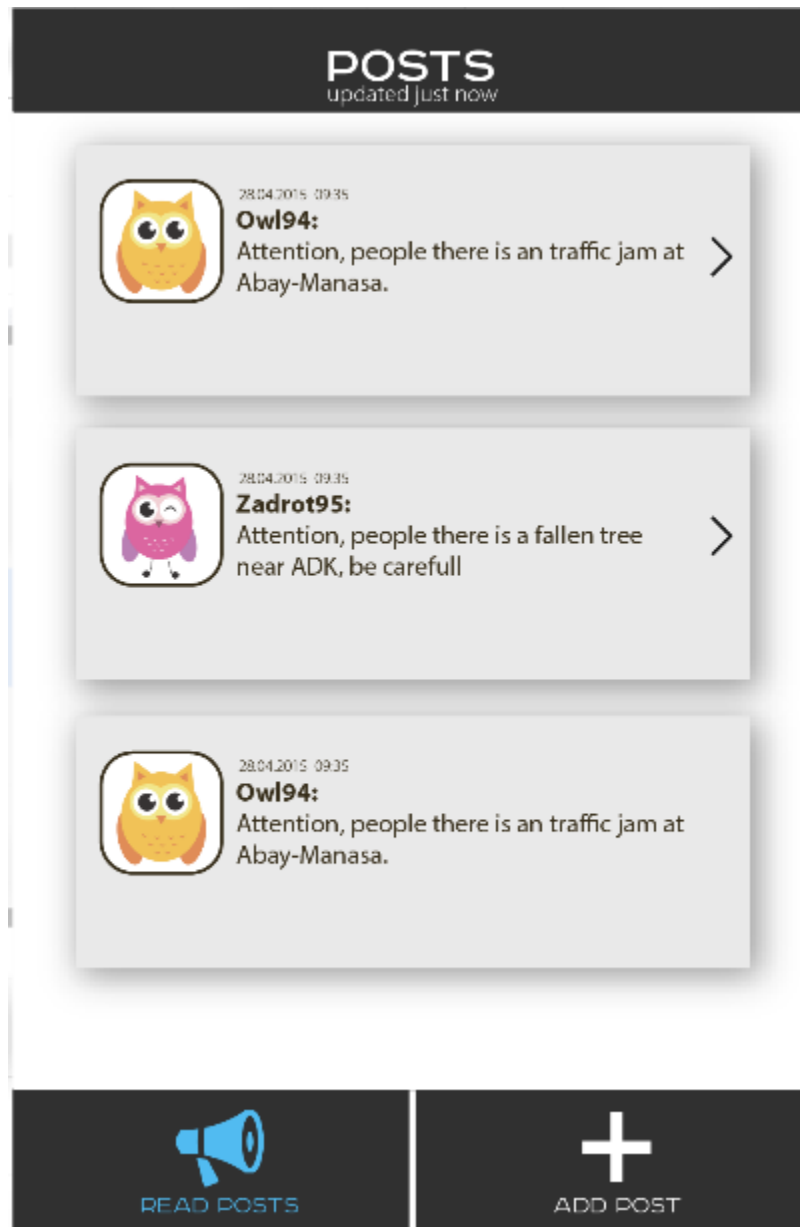


Figure 10. Interface of page "Main Page"

Source: Author

- 2) Add post section: the user can add new post to share urgent information as shown in illustration below (Figure 11).

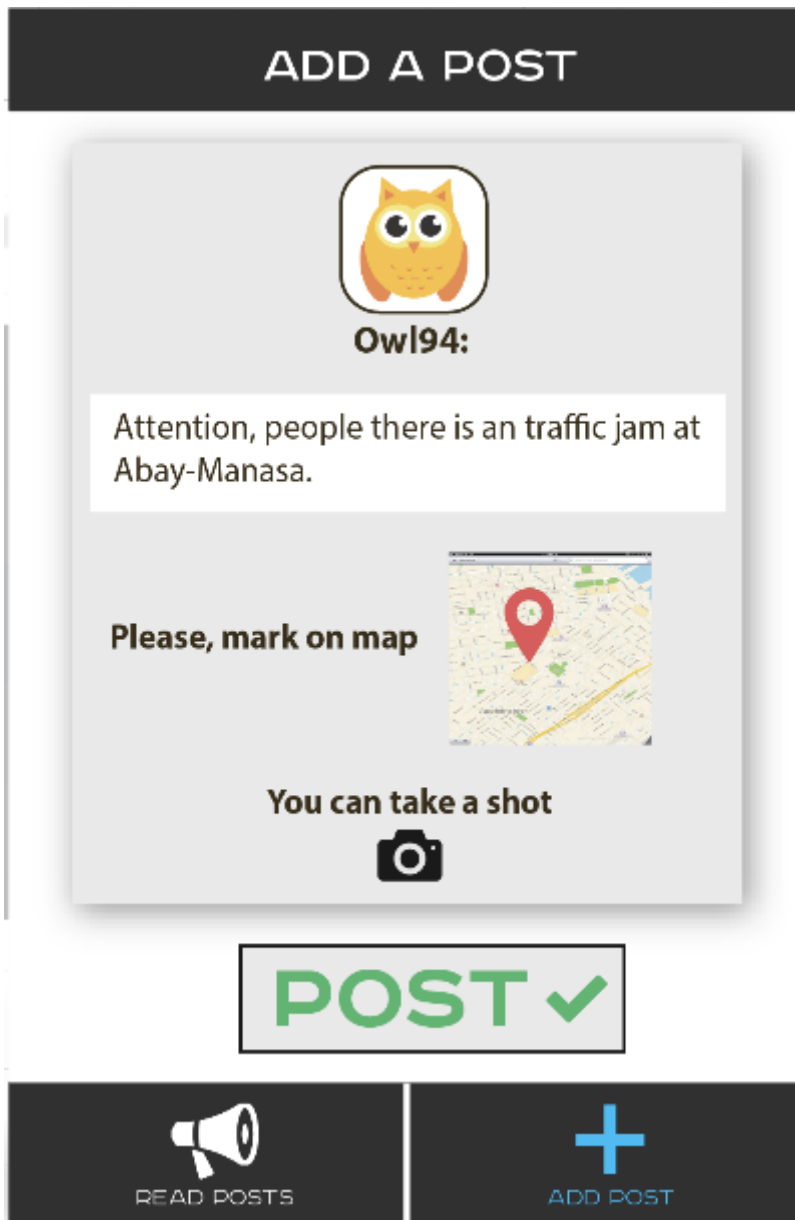


Figure 11. Interface page "Product"

Source: Author

The Interface design has built by all standards of heuristic approaches and also had created new vision and new User Interface Design approach.

5. Results and Discussion

The aim of the diploma project is to create a unique application for sharing urgent information immediately. This data will be available via any phone working on iOS.

5.1. Economic Efficiency

Economic efficiency means getting the maximum possible benefit from available resources. Only rational behaviour will make it in most effective and productive way. Rationality is that how the producer and consumer goods serves in maximum efficiency and to maximize the benefit and minimize costs.

Efficiency of capital investments is one of the most crucial parts in efficiency of economic system. The efficiency of capital investments is equivalent to that rate of rebate which would make the present estimation of the arrangement of annuities given by the profits anticipated from the capital resource amid its life simply equivalent to its supply cost.

Capital investments effectiveness is measured by indicators, which includes the overall effect of capital investment, the rate of return, payback period, the comparative effectiveness and other indicators of economic efficiency of capital investments are used for comparing other alternative investment opportunities and picking the most effective and productive case(project).

Huge role in economic process developer is taking place. It is very important not only to create, but debugging and operating is also crucial.

5.2. App Store (iOS)

The App Store is a digital distribution platform for mobile apps on iOS, developed and maintained by Apple Inc. The service allows users to browse and download applications that are developed with Apple's iOS SDK. The apps can be downloaded directly to an iOS device, or onto a personal computer via iTunes (also developed and maintained by Apple Inc.).

The App Store opened on July 10, 2008, via an update to iTunes. It allowed Apple to control the quality of apps for the first time and to introduce a charge on top of the basic cost of the app. On July 11, the iPhone 3G was launched and came pre-loaded with iOS 2.0.1 with App Store support; new iOS 2.0.1 firmware for iPhone and iPod Touch was also made available via iTunes. As of February 10, 2015, there are at least 1,100,000+ third-party apps officially available on the App Store.

5.3. Calculation of the costs of developing software

In the economic part of thesis is calculating cost covering design work and calculating cost-productiveness of created system. The estimation of spending includes outcomes below:

- cost of materials;
- cost of the basic and additional salary;
- unified social tax;
- cost of maintenance and operation of equipment;
- Additional cash costs.

5.3.1. Calculation of the cost of materials

For developing the application, necessary to have one work place and one workstation for installing the server part of software package. To make the developing part needed one MacBook, in that case MacBook Pro 13 MGX72 with Retina – 23 000 czk.

In the development of the project have had to buy:

- A4 Papers – 500
- Notebooks - 650
- Mouse - 6500
- USB Flash Drive 8GB - 5000
- Books - 4500

- Other unexpected expenditure – 5000

Total: 24 500 . Total expenditure on materials is: 24 500 czk .

5.3.2. Expenditure of salaries

The basic salary is calculated in base of developer and designer's salary (Table 4), for the duration of developing the project.

Category of employees	Quantity	Salary czk./mon.)
Designer	1	9 000
Developer	1	11 000
Total		20 000

Table 4. Staffing of the project team
Source: Author

Approximately the average number of the working days in our country is 22 days. Additional monthly fee charged at the rate (K_{ad}) of 20% of the basic salary is (Formula. 1):

$$S = S_m * K_{ad}$$

It is: $(20\ 000 * 20) / 100 = 4000$ czk.

Social security payroll accrual of is 26% of the basic salary (Formula 2):

$$P = S_m * 26\% = 20\ 000 * 26\% = 5200$$
 czk.

Position	Quantity of workers	Salary czk./mon.	Quantity of worked days	Salary in one day czk./day.	Total czk.
Developer/ designer	1	20 000	60	667	40 000
Additional salary					7800

Accruals to social security payroll	3900
Total:	51 700

Table 5. The calculation of the basic and additional wages.

Source: Author

5.3.3. The costs of operation and maintenance of equipment

Software development was made on MacBook, which averagely costs 20 000 czk.

Cost of 1 hour of notebook time is calculated by the formula:

$$C_t = S_e / T_y * K \quad (\text{Formula. 3})$$

Where, S_e - Annual operating costs;

T_y - Annual Usable PC time;

K - coefficient of utilization of the PC (0.7).

The annual operating cost calculated by the formula:

$$S_e = S_a + S_z \quad (\text{Formula. 4})$$

where S_a - Deductions for depreciation (12.5% cost of the PC);

S_z - The cost of maintenance (2% cost of the PC).

$$S_e = 20\,000 * (12.5 + 2) / 100 = 2900 \text{ czk.}$$

Yearly usable time of hardware is figured as the result of the quantity of working days in a year at the season of the PC in the day):

$$T_y = 252 * 7 = 1764 \text{ hour.}$$

At that point the expense of one hour of PC time (Formula. 5):

$$C_t = S_e / T_y * K$$

$$C = 2900 / 1764 * 0.7 = 1.15 \text{ czk.}$$

The amount of working hours spent on the progression and examining of programming structure is $(66 * 8) = 528$ hours.

In this way, the cost of bolster and operation of the PC are equal to:

$$P = 528 * 12.9 = 607.6 \text{ czk.}$$

Cost item	Sum, czk.
cost of materials	24 500
costs of operation and maintenance of equipment	607.6
Main salary	40 000
Additional salary	7800
Accruals to social security payroll	3900
Total:	76 807.6

Table 6. The estimated total costs of the work

Source: Author

5.3.4. Calculation of profit

In first year application will be spread for nothing to get more acclaimed, and afterward following one year the expense of utilization will be 12.2 czk. In Apple advertise more than 150 million individuals.

$$659075,2/150 = 6663 \text{ people}$$

If 6663 users will download this application all expenses will be shut. On the off chance that in one day 10 clients will download this application: $30*30 = 900$ users in month

$$900 * 12 = 10800 \text{ user in year}$$

In a 8 month 6663 users will download this application and all expenses will be shut, and after that time all downloaded application will make a benefit.

5.4. Labor protection and Industrial Ecology

5.4.1. General information about health and labor protection

The widespread use of computers has allowed to increase productivity of labor, but also raised some problems, including protection of workers from harmful factors associated with the use of computer technology.

According to the theme of the diploma project is necessary to design and realize a Web-based application with e-learning system. This section of the diploma project includes following: analysis of the working conditions, identification the adverse factors, which affect the environment and development of concrete measures to ensure the safety of the worker. For analysis of the working conditions a place of training- Czech University of Life Sciences was chosen. The building of university consists of 4 floors, with a total floor area of 8578 m². For consideration computer room C28 located on the 2nd floor is taken. Total area of the computer room is 30 m², height is 2.5 m and room has two windows. The audience has following items of technical equipment: 20 workplaces, 20 personal computers (20 computer cases, 20 monitors, 20 keyboards and 20 mice), 2 cameras and 1 air conditioning unit.

5.4.2. Analysis of dangerous and harmful factors

University staff can possibly face the impact with physically dangerous and harmful factors, such as increased noise, increased ambient temperature, the absence or lack of natural light, the lack illumination of the workplaces, electric hazards, the risk of a fire, static electricity, electromagnetic radiation and others.

These listed adverse factors might lead to decrease in efficiency, caused by the developed exhaustion.

5.4.3. Measures of norms

In order to ensure the established norms (Table 7) of microclimate parameters and air quality in computer rooms and other spaces ventilation is in use. The design of the

ventilation system involves determining the flow of air for ventilation of the computer room and computer cooling cases.

The room has general and local artificial ventilation. Minimum flow rate is determined based of 50 - 60 m³ / h consumption per person, but not less than two air changes per hour.

Air conditioning is the process of creation and preservation of permanent or alternate set air parameters in the workplace of the production facilities, carried out automatically. The computer room has air conditioning unit.

Period of THE	Microclimate parameters	Value
Winter	Room temperature	22...24°C
	Relative humidity	40...60%
	Velocity of air	till 0,1m/sec
Summer	Room temperature	23...25°C
	Relative humidity	40...60%
	Velocity of air	0,1...0,2m/sec

Table 7. Microclimate parameters

Source: Author

Work of many teachers, staff and students is associated with eye strain, therefore compliance with requirements for lighting of rooms and workplaces of employees is an important element of the rational organization of labor.

In the room there are two light sources - natural and artificial light sources, created by people.

The workplaces of students and teachers working with displays are located away from windows and oriented so the window openings are located on the side. The windows have regulated blinds.

Mainly fluorescent lamps are used for artificial lighting. They have high luminous efficacy (up to 75 lm/W and more), long life period (up to 10000h), low brightness of the illuminated surface, close to the natural spectral composition of the emitted light, which provides good color rendition.

Noise level in the workplace must not exceed 50dBA. Noise reduction is achieved by increasing the insulation envelope, sealing the perimeter of windows, sound proofing points of utility connections passage or crossing with walls.

Protection from noise generated in the workplace internal sources, as well as the noise coming from the outside, is accomplished by means of: reduction of noise at source, the use of collective and individual protection, rational layout and acoustic treatment service facilities.

5.4.4. Elimination of static loadings and monotony of work

For the purpose of decrease in intensity of work and reduction of negative influence of monotony carrying out the following actions is necessary:

- breaks in work;
- change of the contents and rate of the works performed in the free rhythm and in beat imposed by the PC program (alternation of text editing and data input);
- introduction to the mode of work of functional music;

The regime moments of the working day (gymnastics, breaks, action of third-party irritants) are effective only if their character and location strictly consider the corresponding "critical moments" of labor process, to reveal which the analysis of dynamics of working capacity and studying of psychophysiological shifts in an organism of the working allows.

For saving of health of the user it is necessary to adhere to some simple rules:

- the workplace has to be convenient and provide normal functioning of the musculoskeletal device and blood circulation;
- total operating time for the PC during the working day should not exceed 4 hours;
- after each hour of work it is necessary to do a break, at least, for 10-15 minutes during which it is necessary to rise and execute a number of exercises for eyes, a waist, hands and feet;

- to create the external conditions weakening impression of monotony of work (functional color of the production room, the equipment of rooms of psychological unloading, etc.).

5.4.5. Lighting

Spot lighting is not recommended. The general lighting is used. Maximum illumination of 400 lx, brightness less than 15 units, pulsation less than 10%.

Illumination on a table surface in a zone of placement of a brief has to be 300 - 500 lx. Installation of lamps of spot lighting for illumination of documents is allowed. Spot lighting should not create flares on a surface of the screen and increase illumination of the screen more than 300 lx.

It is recommended to use lamps this world, cold this world, the closest to natural light. Power of lamps of 36-40 W, temperature of 3000-4200 degrees Kelvin, then they do not give a high ultraviolet light.

The main stream of natural light has to be at the left. Sunshine and flares should not come into the view working with computer.

5.4.6. Electrical safety

Electrical installations, which include almost all PC hardware, are for a pose great potential danger for a person, as while the operating or conducting maintenance works one can touch parts under high voltage. The specific risk of electrical installations is that the live conductors, computer encasing and other equipment, put under voltage as a result of damage (breakdown) of isolation, do not provide any signals that would warn people of the danger.

When working on electric installations in order to prevent any injuries it is strictly important to implement and comply with the appropriate organizational and technical measures. The organizational measures include: issuing written or verbal work order, work permits, supervision during the operation, registration of the work breaks, job transfers, and end of the work.

In the room to ensure the electrical safety of electrical installations in operation, along with these activities are used technical protection measures, which include: electrical insulation of live parts, protective grounding, neutral grounding, and electrical separation of the network, lowered voltage, double insulation.

5.4.7. Fire safety

Fire is a danger, being associated with large losses. The risk of fire is possible if there are combustible materials, oxidizers and ignition sources in one place.

The fuel component are building materials, windows, doors, flooring, furniture, insulation of power and signal cables, radio parts and connecting cables between cells, blocks, panels, racks, cabinets, structural elements made of plastic materials, liquids for cleaning components and PC assemblies from pollution.

The sources of ignition may be sparks, arcs, and hot spots PC components and assemblies.

Sources of ignition occur in electrical and electronic equipment, devices used for the maintenance of PC components, as well as in air conditioning and heating systems.

One of the most important tasks of fire prevention is to protect building structures from destruction and providing them with adequate strength under high temperatures during a fire. Building structures shall be constructed from brick, concrete, glass, metal and other non-combustible materials. To prevent the spread of fire from one side of the building to another fire barriers in the form of fire walls, partitions, ceilings.

For fire suppression in the initial stage primary extinguishing means are used in CULS: hand-held and portable fire extinguishers, indoor and outdoor fire, etc.

In buildings, fire hydrants are installed on site staircases, at entrances, i.e. in accessible and visible places.

5.4.8. Safety

Ergonomic measures should provide reduction of tiredness by reducing psychological, psycho-physiological stress by providing the optimum parameters of work. The important role is played given to planning of the workplace.

For comfortable work table must satisfy the condition: the height of the work surface is in the recommended range of 68-76 cm.

Great importance is given to the characteristics of the desk chair. Thus, the recommended seat height above floor level is within the 42-55 cm

While working at the computer physicians advise to mount the screen at a distance of 50-60 cm from the eyes.

Great importance is also given to the correct working posture user. Uncomfortable working posture can cause pain sin muscles, joints and tendons. Requirements for working posture by video terminals are as follows:

- ✓ head should not be tilted more than 20 °,
- ✓ shoulders should be relaxed,
- ✓ elbows - at an angle of 80 ° ... 100 °,
- ✓ Forearms and hands - in a horizontal position.

The reason for wrong posture is due to the following factors: there is no good stand for the documents, the keyboard is located too high, and documents - too low, there is no place to put the arms and hands, not enough leg room.

5.4.9. Human impact on the environment of the object and activities for environmental safety

Personal computers pollute the environment not less than cars. Computer de-ionizes the environment and reduces humidity. For the production of a computer with a monitor weighing 24 kg requires 240 kg of fossil fuels, 22kg of chemicals, 1.5 tones of water. The use of fossil fuels contributes to global warming, and many chemicals are toxic. In particular, the chemicals used in the manufacture of computers, circuit boards, monitors, CPUs and endanger the health of employees of factories, as well as affect the drinking water reservoirs around the landfill.

Respiratory diseases are primarily allergic in nature. This is due to the fact that the housing and monitor the system board unit are heated and emit harmful substances in the air, especially if the computer is brand new. The computer creates an electrostatic field around itself which attracts dust and which in turn deposits into the user's lungs.

To solve these problems, you should ensure the proper usage of computers, it is necessary to use materials and bio plastics in the case and components production and minimize the use of hazardous substances.

6. Conclusion

In this diploma project the main goal was the development of mobile application based on iPhone Operating System (iOS). Necessity to develop the such application using capabilities of technology today is the foundation of this project. The specific character of the diploma project is the theme of sharing urgent information among society of the city.

The main steps of diploma project was analysis of the market. It showed that there is no applications for posting immediately information of the situations in the city such as car accidents, traffic jams and any kind of unforeseen circumstances. Therefore, it shows the relevance of the project. Development of the models and realization of the concept of application were the next steps of the project. Then there were the steps of implementation of architecture of application and implementation of design of this mobile application.

By the analysis there no such mobile applications in Czech Republic market and this diploma project result gives such application. It is developed for all categories of users. There are functions like login/registration, write a post, read the posts, pin on a map.

The software product was designed for the iPhone Operating System platform in the programming language Swift. During the implementation of the diploma project the analysis of the methods and modern approaches, market analysis, comparison of technology was conducted. Diploma project includes detailed architecture of application and structure of the interaction, such as procedures, classes, methods, and databases.

Mobile application "Attention" was developed as a result of this diploma project, which will be distributed through an online App Store (iOS).

7. References

1. **Mobile Technology.** Software Development For The Mobile Market. [Online] [Cited: February 4 2018], from <https://www.bbconsult.co.uk/expertise/mobileapplications/mobiletechnology>.
2. **Fitzek Frank H. P., Reichert Frank.** (2007). Mobile phone programming and its application to wireless networking ISBN: 978-1-4020-5968-1
3. **TutorialsPoint.** Distributed Architecture. Cited 14.02.2018, from https://www.tutorialspoint.com/software_architecture_design/distributed_architecture.htm
4. **Karmouch Ahmed, Korba Larry, Madeira, Edmundo R. M.** (2004). Mobility aware technologies and applications : first international workshop, ISBN: 3-540-23423-3
5. **Peter van de Put.** (2013). Professional iOS Programming – ISBN: 978-1-118-66113-0
6. **Stephen G. Kochan.** (2008). Programming in Objective-C 2.0 – ISBN: 978-0-321-56615-7
7. **Parseplatform.com,** 'Parse'. [Online]. [Cited: 25.02.2018.] <http://docs.parseplatform.org/ios/guide>.
8. **Wikipedia.com.,** Extensible Application Markup Language. *en.wikipedia.org*. [Online]. [Cited: 03.03.2018.], from https://en.wikipedia.org/wiki/Extensible_Application_Markup_Language.
9. **Wikipedia.com]** Android OS (operating system). *en.wikipedia.org*. [Online]. [Cited: 05.03.2018.] [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
10. **Wikipedia.com.** iPhone Operation System. *en.wikipedia.org*. [Online]. [Cited: 05.03.2018.] <http://en.wikipedia.org/wiki/IOS>.
11. **Leontev. S,** Ispolzovanie mobilnych telefonov v raznykh stranach. *www.cossa.ru*. [Online]. [Cited: 12.03.2018.] <http://www.cossa.ru/articles/152/37433/>.
12. **Markov E,** (2003), Arhitektura raspredelennykh prilozhenii. *www.pcweek.ru*. [Online]. [Cited: 08.03.2018.] <http://www.pcweek.ru/infrastructure/article/detail.php?ID=66147>.
13. **Polyak-Braginskiy A.V.** (2002), P54B Set svoimi rukami. SPb.: BHV-Peterburg, pp. 320.
14. **Charlez Petzold,** (2010), *Microsoft Press. Mobile Programming* [book auth.] Charlez Petzold . Electronic book (PDF).
15. **Patrice Pelland,** (2011), *Microsoft Press. Moving to Microsoft Visual Studio 2010* [book auth.] Patrice Pelland, Pascal Pare, Ken Haines. Electronic book (PDF).

Appendix

List of helpful sources for developing iOS Applications:

Online courses:

1. **iTunes U:** "Developing iOS 10 Apps with Swift by Stanford"
2. **Udacity:** "Intro to iOS App Development with Swift"
3. **Udemy:** "The Complete iOS 10 & Swift 3 Developer Course"

Software, tools and utilities:

1. **Design:** Sketch
2. **IDE:** AppCode
3. **Dependency Manager:** Cocoapods

Resources and Tutorials:

1. **Apple Developer:** Official resources from Apple
2. **Ray Wenderlich:** iOS Development Tutorials
3. **AppCode:** Free programming tutorials for iOS
4. **Tutorials Point:** iOS (iPhone, iPad) Tutorials

News, forums and blogs:

1. **iOS Dev Weekly:** Podcasts, digests, tools, blogs design by Dave Werner.
2. **Apptractor:** News, tutorials and articles in Russian.
3. **Reddit:** Many readers and interesting threads

Most downloaded mobile applications on iOS:

Top Non-Game Apps by Revenue, Worldwide, Q1 2017





















Overall			App Store		
1		Netflix	1		Netflix
2		LINE	2		Tinder
3		Tinder	3		Pandora
4		Pandora	4		LINE
5		Spotify	5		Spotify
6		LINE Manga	6		YouTube
7		HBO NOW	7		HBO NOW
8		YouTube	8		LINE Manga
9		Google Drive	9		Tencent Video
10		Tencent Video	10		QQ

Figure 12. Top Non-Game Apps by Revenue

Source: <http://news.softpedia.com/news/facebook-and-messenger-are-the-most-downloaded-apps-on-android-and-ios-514933.shtml>